

Transcrição

Continuando com a construção das tabelas de Distribuição de Frequências, conheceremos o método `.crosstab()` de Pandas que faz o **cruzamento** sua [documentação pode ser acessada neste link](#).

Na primeira célula da parte "Método 2" de nosso notebook, teremos um dicionário que transformará o dado numérico em uma `string`, baseando-se

```
sexo = {0: 'Masculino',
        1: 'Feminino'}

cor = {0: 'Indígena',
        2: 'Branca',
        4: 'Preta',
        6: 'Amarela',
        8: 'Parda',
        9: 'Sem declaração'}
```

Construiremos a tabela na célula seguinte, e escreveremos que `frequencia` será igual a `pd.crosstab()` recebendo a variável `Sexo` nas linhas e (

```
frequencia = pd.crosstab(dados.Sexo,
                          dados.Cor)

frequencia
```

Cor	0	2	4	6	8
Sexo					
0	256	22194	5502	235	25063
1	101	9621	2889	117	10862

Os valores da tabela correspondem ao cruzamento das variáveis em questão.

Para visualizarmos melhor, renomearemos a tabela de `frequencia` com `.rename()` recebendo o parâmetro `index` igual ao dicionário `sexo` para

Em seguida, passaremos também o `inplace` igual a `True` para que já modifique diretamente o próprio `DataFrame` que está tratando. Caso contrário somente `frequencia` igual a `frequencia.rename()` recebendo apenas o `index`.

```
frequencia = pd.crosstab(dados.Sexo,
                          dados.Cor)

frequencia.rename(index = sexo, inplace = True)

frequencia
```

Cor	0	2	4	6	8
Sexo					
Masculino	256	22194	5502	235	25063
Feminino	101	9621	2889	117	10862

Faremos a mesma coisa copiando e colando a linha de `.rename()` na sequência. Depois, substituiremos `index` por `columns` para o caso das colunas. O código ficará assim:

```
frequencia = pd.crosstab(dados.Sexo,
                        dados.Cor)
frequencia.rename(index = sexo, inplace = True)
frequencia.rename(columns = cor, inplace = True)
frequencia
```

Cor	Indígena	Branca	Preta	Amarela	Parda
Sexo					
Masculino	256	22194	5502	235	25063
Feminino	101	9621	2889	117	10862

Feito isso, a visualização do cruzamento e as análises dos dados ficarão mais claras.

Em seguida, criaremos a tabela de percentuais. Copiaremos o mesmo bloco de código anterior, e substituiremos `frequencia` por `percentual`.

Dentro do `.crosstab()`, escreveremos `normalize` igual a `True` para normalizarmos os valores. Por fim, multiplicaremos por `100`.

```
percentual = pd.crosstab(dados.Sexo,
                        dados.Cor,
                        normalize = True) * 100
percentual.rename(index = sexo, inplace = True)
percentual.rename(columns = cor, inplace = True)
percentual
```

Cor	Indígena	Branca	Preta	Amarela	Parda
Sexo					
Masculino	0.333160	28.883394	7.160333	0.305830	32.617126
Feminino	0.131442	12.520822	3.759761	0.152264	14.135867

Este método também permite que façamos **agregações** de acordo com uma determinada função usando outra variável dentro do cruzamento.

Por exemplo, copiaremos e colaremos o bloco anterior e substituiremos o parâmetro `normalize` pela função agregadora `aggfunc` igual a `'mean'`.

Neste caso, calcularemos a média da `Renda` que será o parâmetro `values`. Por fim, tiraremos a multiplicação por `100`.

```
percentual = pd.crosstab(dados.Sexo,
                        dados.Cor,
                        aggfunc = 'mean',
                        values = dados.Renda)
percentual.rename(index = sexo, inplace = True)
```

```
percentual.rename(columns = cor, inplace = True)
percentual
```

Cor	Indígena	Branca	Preta	Amarela	Parda
Sexo					
Masculino	1081.710938	2925.744435	1603.861687	4758.251064	1659.577425
Feminino	2464.386139	2109.866750	1134.596400	3027.341880	1176.758516

Como resultado, calcularemos a renda média dentro do cruzamento de `sexo` e `cor`, o que gerará diversas análises do conjunto de dados. Por exem os homens brancos e amarelos possuem a maior média de renda, enquanto as mulheres negras e pardas recebem o menor rendimento, explicitando a gêneros.

A seguir, continuaremos a abordar a Distribuição de Frequências com variáveis quantitativas.